```
SSSSSS  MM   MM  UU   UU  GGGGGG       BBBBB   YY   YY  TTTTTT  EEEEEE  SSSSSS
SS      MMMMMMM  UU   UU  GG           BB  BB   YYYY      TT    EE      SS
SSSSSS  MM   MM  UU   UU  GG   GG      BBBBB     YY       TT    EEEEE   SSSSSS
    SS  MM   MM  UU   UU  GG   GG      BB  BB    YY       TT    EE          SS
SSSSSS  MM   MM  UUUUUU   GGGGGG       BBBBB     YY       TT    EEEEEE  SSSSSS
```

VOLUME 10, NO. 4,5,6,7 AUG/SEPT/OCT/NOV 1993 PRICE 0.75
SINCLAIR MILWAUKEE USER GROUP P.O. BOX 101 BUTLER, WI 53007

If anv articles are copied kindly credit SMUG BYTES and Author.

1993 OFFICERS AND WORKING MEMBERS

All Area Codes ARE 414

President - Bill Heberlein
527-2191

V. Pres - Neal Schultz
353-4522

Secretary - Bud Dankert
251-4464

Treasurer - Dick Cultice
542-3591

Eductation -Dick Cultice
542-3591

Librarians:

Tape/disk - Ed Pawlowski
1-338-0260

Books - Dennis Nickel
1-284-9691

Sub Group Leaders

C - Dick Cultice
Meetings Suspended

Games - Neal Schultz
3rd Sat of Month 353-4522

Hardware - Gordon Kraemer
Call for Information 421-0179

QL - Rudy Hilsmann
3rd WED of Month 251-5291

Spectrum - Rudy Hilsmann
3rd WED of Month 251-5291

SMUG BYTES is published by the MILUWAUEE AREA USERS GROUP--a not-for-profit group devoted to serving the interests of those who own, use and/or are interested in the Timex/Sinclair family of computers.

SMUG maintains a gratis excha~ge of newsletters with approximately 20 Users Groups across the U.S. and Canadea. Clubs not sending newsletters for more than 6 months are removed from our mailing list.

Newsletter subscription is available for only $10/12 issues to non-members or free with a club membership of $20/yr.

Advertizing cost for 1/2 page ad running for 6 issues is $10. The copy may be changed each issue but you must supply copy.

The editor will try to maintain monthly publication but cannot guarantee it and may skip a month from time to time.

Editor and contact person:
Lloyd Dreger (414) 321-0694
2461 S. 79th St.
West Allis, WI 53219

Meetings: 1st Thur. of Month
Place: Equitable Bank
145 & Capitol Drive
Milwaukee.WI
Time: 6:30 to 10:00 p.m.
7:30 Business Meeting

A NEW MEETING DAY

Its now the 1st Thursday of the month. We are not high on the priority list of organizations using the Equitable Bank facilities. As of the beginning of the year we were shifted from our normal 1st Wed of the month to the 1st Tues. We are now being shifted to the 1st Thurs of each month starting in September. Mark it on your calendars as this time they guaranteed it until the end of 1994.

COMMITMENT--WHAT IT TAKES

A good plan is worth 1000 restarts. With the limited space available on the 2068, one must plan carefully as one doesn't have 8 meg of space to play with, only 39826 bytes. A lot of that space will be used for graphic display data. Right off the top we can forget about animation as it takes way too much space.

Role playing and War Strategy games require a world map. A real world has no edges to fall off of. As Columbus said, "Sailing far enough West will get you coming home East." In the gaming world, make believe worlds can be flat without what is called "wrap around". In space adventure worlds, where designers are more atuned to the real universe, worlds generally wrap in both directions. Going West enters you on the East and going too far North brings you back on the map in the South. This isn't actually the situation on a spherical surface but a close enough representation. This simple wrapping eliminates the edges and corner safer havens one wishes to avoid in war simulation games.

If climatic conditions are going to apply to your wrap around world, you have to have a special 180 degree latitude shift wrap at the poles to bring you back on the map from the north rather than jump to the bottom of the map which would be the south pole. Plotting true maps on a flat surface would present headackes of a major kind. A perfectly square map also has too much area at the poles. Making it rectangular doesn't help. This is why Greenland looks bigger than Australia on Mercater projection maps.

How big a map you need depends upon your objective and how much memory can afford. In Conquest, I needed space for a Master map and since each player discovers the world as he explores it, an additional map space for each of the 3 players. Thats 4 maps! At 6812 bytes/screen we would rapidly run out of memory so we must use character maping. For a 64 by 64 map that amounts to 4096 bytes/map or 16384 for the 4 necessary. We also need space for a city list and still more space for units. With 63 cities and 350 units at 7 bytes/(city or unit) or almost 3k more. It took a lo of thinking to come up with what all had to be stored for each unit--what type it is, who owns it, where it is located in coordinates, what is its condition, any orders it may have received, if on board a transport or in a city, has it finished it turn? and if a transport, how many units it has on board and if its loading--that is more than 7 numbers so things had to be crunched. The cities required ownership, coordinates, production amount of time remaining for production, and flightpath coordinate for fighters.

What was to go into the game was mostly decided before the writing of the code was ever begun. Huge blocks of space were already allocated of which about 12k was reserved for the code itself. This is a bit more code than the average game as a good 4K chunk was to go for the map generator The actual code was written in sections of 1. Map generator 2. City verification 3. Automap making 4. Uni generation 5. Unit move 6. Bookkeepin 7. Combat 8. Computer AI combat.

Too many game designers start out wit ambitious plans when starting to writ a game but because of poor planning find themselves starting over too mar times, lose interest and quit. It is best to write everything down as one has a tendency to forget important bits and pieces. Of course, the fi game is the hardest. Many of the techniques carry over from one game t the next. All that you have seen this far in Smug Bytes can be used over and over again with only minor revisions. The same applies to world wrapping routine of this issue.

How much time does it take to develop a game? More than the average person thinks or even wants to commit to. I was once asked, "How much time do you spend on a computer each week?" The truthful answer is up to 40 hours a week when going full bore. Then slacking off to nothing at times. One burns out and has to recharge the batteries as they say. And at times its sheer drive to finish a project as well. Since I have also written several books for the 2068, I will truthfully say that each book took about 2000 hours and so did Conquest.

Of course many hands make light work so that if 2 or more get together several times a week (It has to be more than once a week!) to test code they have written when not together with each person handling a separate area, it can come together quite rapidly. Modern 3rd or 4th generation games are a lot more complex than their ancesters and the amount of time required goes up in proportion so it becomes imperative that you have a team to do it in a reasonable number of months. One can also help the other with tough code problems. And of course you can bounce the nitty gritty details off of someone. Commercial games are written by teams of workers working on them as a full time job (Most programs today are written in C so that they can be transported to other machines.) on a 40 to 56 hour per week situation. Generally one of the people on the team is the idea person who dreams up the details of what to include in the story line. For role playing games one needs a graphic artist to design monsters (and their animation where space permits) with a lot more detail than the 2 color 8X8 pixels of the 2068 and another to design background or scenery with flowing water and waterfalls. Some have music writers as well for that backgroud music. One also must decide on top down graphics or side view graphics, I have a strong preferance for side view graphics in dungeons. One has to allow for a new view each time one changes directions but it is a lot more realistic. Having characters talk to villages etc. requires lots of memory as one has tc store all the words of each converstion somewhere. An interactive talk where you type in a key word which the character then keys off of for an answer is even more energy intensive. Its not that they are so hard to do but that they take memory. Someday you may be writing for a machine with a lot of memory.

Of course each routine is added in modular sequential order so that it can be immediately tested to make sur it does what its supposed to. Many times the first effort doesn't do it so more complex items are added to make it work. Sometimes it gets so ungainly that it is just best to thro it out and start over. (Don't just toss it in the waste basket, but use it as a thinking plan for the next one.) With games, its not supprising if you have as many pages of discarde code as code you use. There would be no way to test 12K or even 2 or 3K of code for errors all at one time as yc wouldn't know what to watch for or what was doing the lockup if you got one. Its little steps one at a time that linked together make for the complex routine. Most programers forget the first lesson of programing --break down that complex routine to simple sections one can handle. Do th simple basics and then add the bells and whistles. Its one complete routir after another that make up a thing sc complex you don't know how it got tha way. Of course that is what computers are good at--helping us keep organize an ever more complex situations.

Commitment. Like everything worthwhil achieving, it takes a lot of it to dc something grand. If one thinks its worth the effort, it amazing how muct can be achieved. Sometimes it feels even better sharing it with others. But then again, commitment is an old fashioned concept that nobody believe in anymore. It was also called good old fashioned American ingenuity for which the US of A was known for. It amazed the rest of the world. But somehow somewhere we lost it. We better find it again--and fast.

2068 Code Bytes--#4

by Lloyd Dreger

Although this routine is used in a game it can be used anywhere a menu is desired. Like the previous routines in this set, much use is made of the Basic routines already in ROM--if we have to waste 16K of memory we might as well take advantage of it.

This is a general menu routine that is called with the following registers set:

B=# of items in menu

C=What column to run the cursor from. The printing will start one column beyond that. Sorry but each menu starts at the top of the screen at line 0. At least you can have left, middle and right menus if you need submenus.

DE= address of menu names -1. (What each line will say.) Menu items must be consecutive in this list and must not be so long as run onto a 2nd line of screen. The last letter of each line must have bit 7 set as a flag to indicate the end--see how this is used in PX. PN and PM also use PX but are not used to print the menu.

PrintMENU prints the menu and calls MoveCURsor which runs the menu by moving the cursor up and down and automatically recycling when at the top or bottom.

SetTV sets the screen to top screen while GetTV returns the screen to the previous setting.

GETINPut is a general input routine that works for both keyboard and joystick.

The routines can be located anywhere. In CONQUEST its located at 60148 and that address is used for the present code.

```
TV EQU 23612 (A FLAG IN BASIC
VARIABLES)
TVF EQU 52359 (SAVE SPACE IN GAME
VARIABLE TABLE)
60148 58,60,92 STV LD A,(TV)
60151 50,135,204    LD (TVF),A
60154 203,135       RES 0,A   FORCE
UPPER SCREEN
60156 50,60,92  ST LD (TV),A
60159 201          RET
60160 58,60,92 STVO LD A,(TV)
60163 50,135,204   LD (TVF),A
60166 203,139      SET 0,A   FORCE
LOWER SCREEN
60168 24,242       JR ST

60170 58,135,204 GTV LD A,(TVF)
60173 50,60,92     LD (TV),A
60176 201          RET
```

60177 to 60184 is another short routine not used here

```
60185 50,94,235 PMENU LD (LINE),A
60188 205,244,234   CALL STV
60191 120           LD A,B
60192 50,93,235     LD (JFLAG),A
60195 197           PUSH BC
60196 213       PP  PUSH DE
60197 197           PUSH BC
60198 62,22         LD A,22  DO AN AT
60200 215           RST 16
60201 58,92,235     LD A,(JFLAG)
60204 71            LD B,A
60205 58,94,235     LD A,(LINE)
60208 128           ADD A,B
60209 193           POP BC
60210 144           SUB B
60211 197           PUSH BC
60212 215           RST 16
60213 193           POP BC
60214 121           LD A,C
60215 197           PUSH BC
60216 215           RST 16
60217 193           POP BC
60218 209           POP DE
60219 197           PUSH BC
60220 205,80,235    CALL PX
60223 193           POP BC
60224 16,226        DJNZ PP
60226 193           POP BC
60227 205,96,235    CALL MCUR
60230 24,194        JR GTV END OF
ROUTINE

60232 205,32,248 PN CALL FINDM+3
60235 24,3          JR PX
60237 205,29,248 PM CALL FINDM
60240 19        PX INC DE
60241 26            LD A,(DE)
60242 203,191       RES 7,A
60244 213           PUSH DE
60245 215           RST 16
60246 209           POP DE
60247 26            LD A,(DE)
60248 203,127       BIT 7,A
60250 40,242        JR Z,PX
60252 201           RET
```

```
60253 0      JFLAG  DEFB 0
60254 0      LINE   DEFB 0
60255 1      COL    DEFB 1
60256 121    MCUR   LD A,C
60257 61            DEC A
60258 50,85,235     LD (COL),A
60261 72            LD C,B
60262 13            DEC C
60263 175           XOR A
60264 71            LD B,A
60265 58,94,235     LD A,(LINE)
60268 50,93,235     LD (JFLAG),A
60271 197  ERASE    PUSH BC
60272 62,22         LD A,22  DO AN AT
WITH JFLAG AND COL
60274 215           RST 16
60275 58,92,235     LD A,(JFLAG)
60278 215           RST 16
60279 58,95,235     LD A,(COL)
60282 215           RST 16
60283 62,32         LD A, BLANK   PRINT
A SPACE
60285 215           RST 16
60286 62,22         LD A,22 NOW DO AN
AT WITH LINE+B AND COL
60288 215           RST 16
60289 193           POP BC
60290 58,94,235     LD A,(LINE)
60293 128           ADD A,B
60294 50,93,235     LD (JFLAG),A
60297 197           PUSH BC
60298 215           RST 16
60299 58,95,235     LD A,(COL)
60302 215           RST 16
60303 62,62         LD A,">"
60305 215           RST 16 CURSUR
REPRINTED
60306 193           POP BC
60307 205,184,235 CJ CALL GETINP THIS
ROUTINE WAITS FOR AN INPUT WHICH IS
RETURNED IN A. WE WILL ONLY BE
INTERESTED IN THE JOYSTICK UP DOWN AND
BUTTON COMMANDS.
60310 203,79        BIT 1,A
60312 40,11         JR Z, UPA
60314 121           LD A,C
60315 184           CP B
60316 40,3          JR Z,UP1
60318 4             INC B
60319 24,206        JR ERASE
60321 6,0     UP1   LD B,0
60323 24,202        JR ERASE
60325 203,71  UPX   BIT 0,A
60327 40,10         JR Z,BUT
60329 175           XOR A
60330 184           CP B
60331 40,3          JR Z,DW1
60333 5             DEC B
60334 24,191        JR ERASE
60336 65       DW1  LD B,C
60337 24,188        JR ERASE
60339 203,127 BUT   BIT 7,A
60341 40,220        JR Z,CJ
60343 201           RET    ONLY RETURN
IF BUTTON IS PRESSED. YOUR RETURN NOW
MUST TAKE THE VALUE OF B WHICH GIVES
THE LINE NUMBER (STARTING WITH 0) AND
EXECUTE THE COMMAND RELATED TO THAT
SELECTED MENU ITEM.
```

THIS INPUT ROUTINE DEPENDS UPON THE BASIC RST 56 NOT BEING DISABLED SO THE KEYBOARD CAN BE DECIPHERED. SEE THE DISCUSSION IN CODE BYTES #1.

```
60344 175  GETINP   XOR A
60345 50,8,92       LD (LASTK),A  RESET
LASTK TO GET RID OF ANY OLD INPUT
60348 118           HALT  DEBOUNCE
60349 118           HALT
60350 118    JL     HALT
60351 62,14         LD A,14  CHECK
JOYSTICK FIRST
60353 211,245       OUT (245),A
60355 62,3          LD A,3   CHECK BOTH
JOYSTICKS
60357 219,246       IN A,(246)
60359 238,255       XOR 255 INPUT IS
ACTIVE LOW. WE WANT ACTIVE HIGH SO
INVERT.
60361 32,7          JR Z,JM  FOUND NO
JOYSTICK INPUT SO CHECK KEYBOARD.
60363 58,8,92       LD A,(LASTK)
60366 254,0         CP 0
60368 40,236        JR Z,JL FOUND
NOTHING SO CHECK AGAIN.
60370 201           RET
```

THIS INPUT ROUTINE IS VERY FAST. IF YOU WANT MORE WAITING ADD MORE HALTS.

If each of your menu items is a jump to another routine without the need for a return, you can make a table of all the starting addresses and then do this bit of code which is much shorter than all the compares and jump rel.:

```
      LD HL,TABLE-2
      INC B     (B may be zero)
  XYZ INC HL
      INC HL
      DJNZ XYZ
      JP (HL)
   TABLE...addr of first item.
```

If you need to return to the menu for all routines, (which is not the case if you want to return to BASIC) you can push the return address on the stack and leave it there.

Of Mice and Men

New scientific disciplines and endeavors have a big advantage over well established ones in that there are no set guidelines for how anything is done or accomplished. The encrustation of age come later.

For example, back in the dawn of the computering era when programs were hand wired as were memories, the byte was only 3 bits long and counting in octal was in vogue. But a byte couldn't hold enough numbers for a decade so a 4th bit was added creating hexadecimal counting with the very original? use of the letters A through F to represent the notations of the tenth to fifteenth digits. This new 4 bit byte was called the "nybble". Special commands were added to handle decimal notation in the CPU. Soon two nybbles were added to create the 8 bit byte so that it could hold both a high and a low decimal number. Assembly languages were unheard of and one didn't waste memory storing the program. Logic gates were separate devices.

It didn't take very long for the byte to evolve this far. Everything was in a state of flux. Everything was growing rapidly with new ideas coming thick and fast. Things were really moving. Machines were proliferating all over the place, each with its own protocols. A person could speak his thoughts and someone would listen. Or one could write an article and get it published.

But how much has the byte evolved since? None at all. The idea was frozen for all time. So also were a lot of other things. Computering hardware was "getting set in its ways". Nowhere along the way did a committee meet and say, "This is the best way to do things so let's make it a standard." as was the wont of electrical engineers. But it didn't take long. Just a decade or two. If one tried changes now or even spoke of them one would be the "voice of one crying in the wilderness." Right or wrong, protocols and standards have a tendency to stop experimentation and prevent further evolution.

Some of this is necessary to bring sense to the helter skelter, everyone on his own, or worse, "Our captive customers will do it this way just so its NOT compatible with company Z's product." In a certain sense it was the leader of the pack setting the standards which everyone tried to copy that brought us the "clones". But when the leader went her (now days everything is feminine gender!) own way to try to captivate customers, nobody followed.

The computer industry is typical of the rest of science in general only on a compressed time scale. Whereas Physics or Chemistry would take a century or two to evolve to a particular point, computing did it in 1/3rd the time. The sifting and winnowing is much faster. It is now the time of the genetists.

To get papers published in a prestigious scientific journal requires "peer review". First, to even get up for review you have to be somebody of note or come from an institution of note. Being a professor or worse, a student from a 2nd or 3rd rate college won't even get your paper up for review. Once in for review it had better be quite close to the thinking of the people reviewing it--your peers, if it is going to pass muster. Generally, it doesn't make it without a few revisions. If its too far out of line it will be rejected outright unless you have a guardian sponsor guiding it through and backing you all the way on the review committee. Thus, our present method of publication has a strong tendency to continue the status quo (mice, oft times blind) and stifle new ideas (men).

New, radical ideas in an established

field most times get published by younger professionals in obscure journals that the mainstream of the industry or profession never read. As such, great ideas get buried only to be discovered decades and sometimes a century later (some of the stuff in mathematics has been around that long before getting re-discovered and finally put to use). Some of the fault of this is in the rashness of the young men doing the writing. A less radical approach with moderation would have won the day. So there is something to say about the temperance and wisdom? of age. Let's just say that the present method of publication doesn't promote new ideas as well as it should.

One branch of science in particular seems to have produced a fossil--namely astronomy. The Big Bang theory has been around since the '20s and has been patched and repatched so often and still is so full of holes (unexplained facts) it resembles a teenagers favorite jeans (books have been written on what is all wrong with the theory!). Its only saving grace is the explaination of the "red shift" of starlight. To some astronomers this has turned into the "Doppler Red Shift Law". A scientific law means that it has been proven beyond any shadow of doubt. Most science today is just in the "theory" stage...one or two big steps removed from being a law.

The main endeavor today is the search for the Theory of Everything--one grand set of equations that explains the entire cosmos from 10^{-40} to 10^{+40} meters, past, present and furture--essentially combining the Unified theory of quantum mechanics (still missing gravity, and still hunting for missing quarks, gluons and the elusive gravitron) with the Big Bang. Then for all practical purposes Physics will be a completed science. Once again, as in the 1880's Physics will be lamenting its demise.

I find this attempt quite premature. We are a far ways away from being able to write the final equations of a theory of everything especially when one considers the language we have to write this theory in (mathematics) is still clumsy and very incomplete. (Over 90% of the differential equations have no solutions!) The string theories of mathematics still must be worked out. Theoretists sometimes have to invent their own mathematics just like Newton (fluxions of calculus) and Einstein (statistical calculus). Or, if you are religious, "Who do you think you are, God?" After all, this theory will explain creation and Armegeddon (not the battle but the end of the universe)!

But then on the other hand the easiest office that one can set up would be one for theoretical work. All you need is a room and a personal computer with a word processor. After all, you don't have to "prove" anything--just write.

Want proof? Take "black holes"--stars with infinate density. They are black because not even light can escape. Well, infinite density means zero occupied space which means no radius and infers that it has no spin momentum. One big gigantic black hole is reponsible for the start of the Big Bang in the first place. BUT, these holes can radiate energy! Whoops, I though black holes were so strong in gravity that not even light can escape. Well, energy means light, be it in the untra-violet, visible, infra-red, heat waves or radio waves. Its all light and its all quantized. Tiny black holes are so hot that they can radiate away energy even though normal temperature? energy can't escape.

BULL! Energy doesn't have temperature. If a photon gets too energetic it turns this energy into mass. Do you get the feeling something is wrong here? Well, a theory doesn't have to make sense, just sound scientific.

2068 Code Bytes--#5

A routine to print a map that wraps in both directions, i.e., go off the map Westward puts you on the eastern edge, northward on the southern edge or vise versa. The map itself is 64x64. We have two variables called Mapcox and Mapcoy (stored at 52396/7) which define what map coordinates (starting with 1,1) is being printed in the upper left corner of the screen. Since this routine also prints the player maps, it uses the variable Player (at 31539). The master map is stored at 32000. Each map requires 4096 bytes of space so player 1's map starts at 36096, etc. We need a special variable called WRAPF (at 52389). The routine is entered at 60475 preceded by PMO which is a routine that finds the correct space in momory to start from.

```
60442 120       PMO  LD A,B
  BC=MAPCOX,MAPCOY
60443 61             DEC A
60444 38,0           LD H,0
60446 167            AND A
60447 23             RLA
60448 23             RLA
60449 23             RLA
60450 203,20         RL H
60452 23             RLA
60453 203,20         RL H
60455 23             RLA
60456 203,20         RL H
60458 111            LD L,A
THE RIGHT STARTING POINT OF ANY MAP
NOW FIND OUT WHOS MAP
60459 58,51,123      LD A,(PLAYER)
60462 167            AND A
60463 23             RLA
60464 23             RLA
60465 23             RLA
60466 23             RLA
60467 132            ADD A,H
60468 198,125        ADD A,125
ADD THE OFFSET TO MAP ZERO (MASTER)
60470 103            LD H,A
60471 121            LD A,C
60472 61             DEC A
60473 133            ADD A,L
60474 111            LD L,A
60475 201            RET
60476 237,75,172,204 PMAP LD
BC,(MAPCOY)
60480 205,26,236     CALL PMO
60483 175            XOR A
60484 50,165,204     LD (WRAPF),A
RESET WRAP FLAG
60487 229                PUSH HL
SAVE HL ADDRESS FOR CALC. ATTR
60488 17,0,64            LD DE,16384
SET DE TO TOP OF SCREEN
60491 6,3                LD B,3
WE ARE GOING TO PRINT ALL 24 LINES O
THE SCREEN, NOT JUST THE FIRST 21
60493 197      NEXT  PUSH BC :SAVE COUNT
60494 14,8           LD C,8 :8 LINES
60496 6,32     NEXT1 LD B,32 :32/LINE
60498 126      NEXT2 LD A,(HL)
GET USER DEFINED CHARACTER
60499 205,119,237    CALL PRUDG
PRUDG CONVERTS THE CHARACTER CODE TO
THE CORRECT SPOT IN THE UDG TABLE AND
PRINTS IT TO THE SCREEN
60502 205,232,236    CALL WRAP
THIS SUBROUTINE DOES THE ACTUAL
DETERMINATION OF WHEN WE ARE AT A SEAM
AND ADJUSTS HL  ACCORDINGLY. A=WRAPF,
0=NO WRAP, 255=WRAP
60505 254,255        CP 255
60507 32,8           JR NZ,PM1
60509 213            PUSH DE
SAVE POINT ON SCREEN
60510 17,64,0        LD DE,64
60513 167            AND A
60514 237,82         SBC HL,DE
60516 209            POP DE
60517 35       PM1   INC HL
NOW RESET DE
60518 122            LD A,D
60519 214,8          SUB 8
60521 87             LD D,A
60522 19             INC DE
60523 16,229         DJNZ NEXT2
60525 213            PUSH DE
60526 58,165,204     LD A,(WRAPF)
60529 254,0          CP 0
60531 40,5           JR Z,AD-3
AT END OF LINE WE HAVE TO ADD EITHER
96 OR 32 TO HL
60533 17,96,0        LD DE,96
60536 24,3           JR AD
60538 17,32,0        LD DE,32
60541 25       AD    ADD HL,DE
60542 58,51,123      LD A,(PLAYER)
60545 167            AND A
60546 23             RLA
60547 23             RLA
60548 23             RLA
60549 23             RLA
60550 198,141        ADD A,141
60552 87             LD D,A
60553 124            LD A,H
60554 186            CP D
60555 32,3           JR NZ,PM2
RESET TO TOP OF MAP
60557 214,16         SUB 16
```

```
60559 103               LD H,A
60560 209          PM2  POP DE
60561 13                DEC C
60562 32,188            JR NZ,NEXT1
ADJUST DE TO NEXT SET OF 8
60564 122               LD A,D
60565 198,7             ADD A,7
60567 87                LD D,A
60568 193               POP BC
60569 16,178            DJNZ NEXT
NOW DO ATTRIBUTES
60571 225               POP HL
60572 17,0,88           LD DE,22528
60575 14,24             LD C,24
60577 6,32      NEXT3   LD B,32
60579 197       NEXT4   PUSH BC
60580 126               LD A,(HL)
60581 205,254,236       CALL FINDAT
THIS ROUTINE CALCULATES BOTH INK AND
PAPER FOR THE CHARACTER AND STORES IT
IN A
60584 18                LD (DE),A
60585 205,232,236       CALL WRAP
60588 254,255           CP 255
60590 32,8              JR NZ,NIX
60592 213               PUSH DE
60593 17,64,0           LD DE,64
60596 167               AND A
60597 237,82            SBC HL,DE
60599 209               POP DE
60600 19        NIX     INC DE
60601 35                INC HL
60602 193               POP BC
60603 16,230            DJNZ NEXT4
60605 213               PUSH DE
60606 58,165,204        LD A,(WRAPF)
60609 254,0             CP 0
60611 40,5              JR Z,A1-3
60613 17,96,0           LD DE,96
60616 24,3              JR A1
60618 17,32,0           LD DE,32
60621 25           A1   ADD HL,DE
60622 58,51,123         LD A,(PLAYER)
60625 167               AND A
60626 23                RLA
60627 23                RLA
60628 23                RLA
60629 23                RLA
60630 198,141           ADD A,141
60632 87                LD D,A
60633 124               LD A,H
60634 186               CP D
60635 32,3              JR NZ, PR3
60637 216,16            SUB 16
60639 103               LD H,A
60640 209       PR3     POP DE
60641 13                DEC C
60642 32,189            JR NZ,NEXT3
60644 201               RET
```

```
60645 125    WRAP    LD A,L
60646 254,63         CP 63
NOTE: 63, 127, 191 AND 255 FOR L
INDICATE THE EASTERN EDGE OF THE MAP
60648 40,11          JR Z,CONT
60650 254,127        CP 127
60652 40,7           JR Z,CONT
60654 254,191        CP 191
60656 40,3           JR Z,CONT
60658 254,255        CP 255
60660 192            RET NZ
60661 62,255   CONT  LD A,255
60663 50,165,204     LD (WRAPF),A
60666 201            RET
```

The Findat routine depends upon how you set the various UDG characters on your map. For Conquest they are: 0 blank, 1 sea, 2 land, 3 mountain, 4 shoal, 5 city, 6-15 units whose color will be yellow, red and white for players 1 to 3 respectively. For cities and units ownership is designated in the top 2 bits.

```
60667 254,0 FINDAT CP 0
60669 32,3         JR NZ,AT1
60671 62,7         LD A,7    P=BK,I=W
60673 201          RET
60674 254,1   AT1  CP 1
60676 32,3         JR NZ,AT2
60678 62,15        LD A,15   P=BL,I=W
60680 201          RET
60681 254,4   AT2  CP 4
60683 48,3         JR NC,AT4
60685 62,32        LD A,32   P=GR,I=BK
60687 201          RET
60688 254,4   AT4  CP 4
60690 32,3         JR NZ,AT5
60692 62,41        LD A,41   P=AQ,I=BL
60694 201          RET
60695 254,5   AT5  CP 5
60697 32,3         JR NZ,C2
60699 62,59        LD A,59   P=W,I=MG
60701 201          RET
60702 254,69  C2   CP 69  PLAYER1?
60704 32,3         JR NZ,C3
60706 62,48        LD A,48   P=BK,I=Y
60708 201          RET
60709 254,133 C3   CP 133 PLAYER2?
60711 32,3         JR NZ,C4
60713 62,16        LD A,16   P=BK,I=R
60715 201          RET
60716 254,197      CP 197 PLAYER3?
60718 32,3         JR NZ,P1
60720 62,56        LD A,56   P=BK,I=W
60721 201          RET
```

We are runnimg out of space so we wil finish this routine in the next installment.

Brave New World and the Good Old Days

With the Graying of America most of us have a tendency to be on the elderly side and can reminisce about the good old days. Each of us have fond memories of our youth as the mind tends to remember only nice things and wipes out painful ones. Anyone over 40 is out of college about 20 years which puts those days in the pre-home or personnel computer age. If you were an engineer or a scientist you were proficient in operating the slip stick better known as the slide rule. They were our constant companions and almost a symbol of our profession. The electronic calculator was just coming into its own but at $250 was a bit too pricy for a poor student. For more precision we had the old electrically driven mechanical calculators to grind out multiplication and division--and I do mean grind out. When you got your first job, and everyone was hired in those days, you got yourself a battery operated calculator.

Typewriters were in vogue as well and you were really living high on the hog if you owned an electric one. Erasing a typo so it didn't show was a real art--else it was retype the entire page. No spelling checkers and hence no malaprops. (Mrs. Malaprop was a character in a Restoration Comedy by Sheridan who had a facility for inserting the wrong word in a sentence-- like infinate when infantile was meant. In print, there for their-but not in speaking.) In this sense the good old days were better because our present younger generation seems to have never been trained in good English useage. Its bad when college grads never even heard about a split infinitive much less a dangling participle. Omission of words in sentences and awkward sentence structure is rampant in prominent National professional magazines.

Then of course we have the buzz words of every technology. These are used excessively by individuals who think they are impressing their audiences when in fact they have lost them. We have inactive and active and even super active and reactive so what is the need for proactive? If we have proactive, is there conactive? What is the difference between an astronaut and a cosmonaut?

Even worse, and NASA is the biggest abuser, is the use of acronyms. I have no objection if the user defines each one the first time it is used thus informing his (the sexually neutral issue is another) reader. But too many make it difficult to comprehend. Take the MQZ and insert it into the TVS which is then given to PVR for SDS means about as much as:

The time has come, the walrus said,
To speak of many things--
Of ships and shoes and sealing wax,
Of cabbages and kings.
Of why the sea is boiling hot,
And weather frogs have wings.
--Lewis Carrol
Through the Looking Glass

If you really want to make yourself unintelligible use 2 or 3 5 syllable words for every single syllable word. I was appalled to hear that AT&T at one time was actually advised its speakers to raise their obscuration index to the graduate college level in an attitude of "If you can't impress them with your technology, overwealm them with your bullshit."

In the good old days we were pollutng all over the place but weren't aware of it. We were also using chemicals that weren't too safe. We are just becoming aware of some of these hazards and more are sure to make an appearance in the future. Of course we have now developed a group of people who make a profit on scare tactics. They banter their select cause to the detriment of everything else and never feel it necessary to present an alternative. Take the nuclear power plant cause. They won't give up the use of electricity and insist that coal burning plants will suffice. Of course, the coal burning plants cause carbon dioxide level increases and a greenhouse effect. Passenger cars are restricted to the nth degree whereas 18 wheelers can spew it out by the ton--if you don't think deisel exhaust is bad for you, get a little too close behind one on an expressway sometime and open a window. Asbestos is no hazard if you don't go around playing with and banging on it. Yet we have spent billions getting it completely out of public buildings. The next big push is lead paint. Prior to WWII almost all paint contained lead. It also is no hazard unless you turn it into dust and inhale it or eat it. Of course we go around tasting paint whereever we go, don't we?

One of the new fads in the Brave New World is computer modeling. Everything from long range weather to the big bang. Most of these models are based on one simplification stacked on another ad infinitum. The users of these theories feel that they are adequately correct and accurate. In reality, old wives tales could do almost as well. Yet because these models spout the latest technology they gain an aura of superiority and accuracy they don't deserve. Putting too much reliance on these over-simplifications and extrapolations can lead to wrong conclusions. Some of the climate models couldn't predict the effect of a volcanic erruption which spewed millions of tons of sulfur dioxide into the upper atomosphere. The results are also all over the place on the effects of the ozone holes. Prediction of earthquakes is also in its infancy.

But modeling isn't the only area of high tech science in need of more substantiation. In delving into some of the basic principals of modern science I was appalled at the lack of and inaccuracy of some of the data supporting some cherished theories. Some very very simple questions havn't any answers at all. Some of these questions have been around since the 1930's which precedes my college education by a few decades. How much credulency should one place in a theory that can only explain selected events and is far from being universal? Looking at the "other" events would disprove the theory.

If we say that mathematics is a type of science, we have some basic inadequacies that have been around for 2000 years. After all these years we still can't adequately take the squar root of a negative number-- that i, representing the square root of -1, says it can't be done. Modern representation of complex numbers consisting of a real part on the axis line of real numbers and an imaginar part off this axis is at best quite artificial. The arithmetic operations of complex numbers limits adding and subtracting only the real parts together and the imaginary parts together. Multiplication gives cross terms and division is nigh to being impossible.

Our brave new world is far from being perfect and complete. There are limitations and impossibilities which our latest theories still are not abl to solve. A lot of people have spent considerable time and effort through the years trying to get answers to some of these remaining basic probler only to give up in the end and go on to other things that have more promi of success. Some of these old chestnuts are really going to requir a whole reworking of some principals and theories be they in a science or in mathematics. There is still a lot of imperfection out there least anyo think otherwise.

THE WALL STREET JOURNAL

"We used to use them when I was your age . . . it's called a pencil."

ENGINEERS AND SCIENTIST WILL ALSO REMEMBER THE SLIDE RULE.

2068 Code Bytes--#6

Correction: In last month's routine (issue #5) kindly insert the following 3 bytes of code after address 60457 shoving everything else down 3 bytes:

```
60458 23              RLA
60459 203,20          RL H
```

The entry address FOR THE PRINT MAP routine than becomes 60479.

We now finish our routine to print a wrapping world map to the screen. There also were 2 address counting errors as well so the correct addresses are as follows:

```
60726 230,192   P1  AND 192
WE ONLY HAVE UNITS LEFT SO WE CAN WORK
WITH ONLY THE TOP TO OWER BITS.
60728 254,64          CP 64
60730 32,6            JR NZ,P2
60732 205,82,237      CALL PAPER
60735 198,6           ~~ADD A,6~~    ADD Y
INK
60737 201             RET
60738 254,128    P2   CP 128
60740 32,6            JR NZ,P3
60742 205,82,237      CALL PAPER
60745 198,2           ADD A,2   ADD k
INK
60747 201             RET
60748 205,62,237 P3   CALL PAPER
60751 198,7           ADD A,7
60753 201             RET
```

We want the background color of land or sea to show around the unit.

```
60754 213    PAPER    PUSH DE
60755 229             PUSH HL
60756 58,51,123       LD A,(PLAYER)
60759 167             AND A
60760 23              RLA
60761 23              RLA
60762 23              RLA
60763 23              RLA
60764 87              LD D,A
60765 30,0            LD E,0
60767 237,87          SBC HL,DE
60769 126             LD A,(HL)
60770 254,1           CP 1       SEA?
60772 32,4            JR NZ,PA2
60774 62,8            LD A,8
60776 24,10           JR PAOUT
60778 254,4           CP 4       MT?
60780 48,4            JR NC,PA3
60782 62,32           LD A,32   P=G
60784 24,2            JR PAOUT
60786 62,40    PA3    LD A,40   P=AQ
60788 225      PAOUT  POP HL
60789 209             POP DE
60790 201             RET

60791 197 PRUDG       PUSH BC
60792 229             PUSH HL
60793 33,88,255       LD HL,65368
60796 230,15          AND 15
60798 23              RLA
60799 23              RLA
60800 23              RLA
60801 133             ADD A,L
60802 111             LD L,A
60803 6,8             LD B,8
60805 126      UDG    LD A,(HL)
60806 18              LD (DE),A
60807 35              INC HL
60808 20              INC D
60809 16,250          DJNZ UDG
60811 225             POP HL
60812 193             POP BC
60813 201             RET
```

This finishes the print a double wrapping map routine. Of course you may want different graphics so may have to write your own attribute routine. You can also have more tha 21 UDG's with this routine and can the pixels anywhere you like merely b changing the address at 60793. In fac we will be doing just that when we print the coordinates for the map wit the following routine.

By making the coordinates a separate routine, we can choose to print them or not. In the editor, where we need human help (or chew up a lot more space with code) we don't want the viewer to know where on the map we are. Whereas, in the game coordinates are a definate help. But we don't wan to take too much screen. So we decide to have both numbers of the coordinates take but one character space. Each number is only 3 pixels wide. Using only even numbers makes them easier to read and requires only half as many. As it is, each of 32 coordinate characters is going to require 8 bytes. They will go down th left side and across line 21 (Basic designations).

```
60814 58,173,204 Prcoor LD A,(MAPCOX)
THE STORAGE PLACE FOR THE UPPER LEFT
COORDINATE OF THE SCREEN
```

```
60817 17,0,64            LD DE,16384
(0,0 OF THE SCREEN)
60820 33,62,208          LD HL,COORD
WHERE 00 FIRST PIXEL IS STORED
60823 6,12               LD B,12
60825 167                AND A
60826 31                 RRA
 DIVIDE COORDINATES BY 2
60827 48,16              JR NC,EVEN
60829 60                 INC A
SET A TO NEXT HIGHER NUMBER
60830 245    AGAIN       PUSH AF
60831 175                XOR A
60832 205,0,238          CALL PRCO
 PRINT A BLANK
60835 123                LD A,E
 ADJST DE TO NEXT PRINT POSITION
60836 198,32             LD A,32
60838 95                 LD A,E
60839 254,0              CP 0
60841 204,251,237        CALL Z,SEAM
60844 241                POP AF
60845 205,0,238 EVEN CALL PRCO
60848 60                 INC A
60849 254,33             CP 33
 AT END OF COOR?
60851 32,2               JR NZ,NOTEND
60853 62,1               LD A,1
 RESET TO FIRST
60855 245  NOTEND        PUSH AF
60856 123                AND A,E
60857 198,32             ADD A,32
60859 95                 LD E,A
60860 254,0              CP 0
60862 204,231,237        CALL Z,SEAM
60865 241                POP AF
60866 16,218             DJNZ AGAIN
 NOW DO ACROSS THE BOTTOM
60868 17,192,80          LD DE,20672
60871 58,192,204         LD A,(MAPCOY)
 GET THE Y COORDINATE
60874 6,16               LD B,16
60876 167                AND A
60877 31                 RRA
 AGAIN, DIVIDE BY 2
60878 48,8               JR NC,EVEN1
60880 60                 INC A
60881 245      AGA       PUSH AF
60882 175                XOR A
 PRINT A BLANK
60883 205,0,248          CALL PRCO
60886 19                 INC DE
 ADJUST DE TO NEXT PRINT POSITION
60887 241                POP AF
60888 205,0,238 EVEN1 CALL PRCO
60891 19                 INC DE
60092 60                 INC A
60893 254,33             CP 33
 TIME TO WRAP?
60895 32,2               JR NZ,NOTE
```

```
60897 62,1                LD A,1
60899 16,236  NOTE        DJNZ AGA
 NOW DO ATTRIBUTES BK INK ON W PAPER
60901 33,0,88             LD HL,ATTRO
50904 62,56               LD A,56
50906 6,22                LD B,22
50908 17,32,0             LD DE,32
50911 110         PC4     LD (HL),A
50912 167                 AND A
50913 25                  ADD HL,DE
50914 16,251              DJNZ PC4
50916 6,32                LD B,32
50918 119        PC5      LD (HL),A
50919 35                  INC HL
50920 16,252              DJNZ PC5
50922 201                 RET
50923 122      SEAM       LD A,D
50924 198,8               ADD A,8
50926 87                  LD D,A
50927 201                 RET
50928 229      PRCO       PUSH HL
50929 245                 PUSH AF
50930 197                 PUSH BC
50931 213                 PUSH DE
50932 254,32              CP 32
50934 40,24               JR Z,ADH
50936 167                 AND A
50937 23                  RRA
50938 23                  RRA
40939 23                  RRA
40940 22,0                LD D,0
40942 95                  LD E,A
40943 167                 AND A
40944 25                  ADD HL,DE
40945 209                 POP DE
40946 213                 PUSH DE
40947 6,8            X1   LD B,8
40949 126       PRCO1     LD A,(HL)
40950 35                  INC HL
40951 20                  INC D
40952 16,250              DJNZ PRCO1
40954 209                 POP DE
40955 193                 POP BC
40956 241                 POP AF
40957 225                 POP HL
40958 201                 RET
40959 36        ADH       INC H
40960 24,240              JR X1
```

Our map is now complete with coordinates. How about being able to sceoll the map in any direction one character at a time? We still have t add a cursor which shouldn't go off screen or across the coordinates whi will be the topic of the next issue.

The Future of Science and Mathematics

It was the best of times.
It was the worst of times.

So begins Charles Dickens classic novel about the French Revelution--A Tale of Two Cities. But so also is the case with present day Science and Mathematics. It has never been so good--or so bad.

Science theorist, especially those working in cosmology, have never had more help modeling their mathematical theories than they do at the present time with the coming of age of the personnal computer also known as the PC. These PC's whip through calculations and draw pictures with astounding speed far exceeding that possible with the mechanical and electronic calculators one had to use 30 years ago. Back then, one slaved away with these slow monsters doing one addition, subtraction, multiplication or division at a time. Errors in entering data into the machine were the biggest concern of the time. 25 years ago you entered your Fortran program into your mainframe computer with a stack of punched cards and got your answers back on another stack of punched cards or a paper printout. Computers didn't have screens, just blinking lights. You were lucky to have 16k of memory as most only had 8k or less. CPU's were all 8 bit.

Maybe we have too much of a good thing. Now almost anybody can set up a 'theoretical lab' as all one needs is a PC. Then one can grind out article after article from the modeling that you do. You don't even need a secretary anymore as everyone can use the Columbus method of typing--. discover and land.

This is the problem. Even with peer review of articles by experts in the field before they are published, too many are getting through without being fully verified or compared to facts. These articles all too often present their speculations as established facts rather than the probabilities that they really are. One of my pet examples is the red shift of starlight which is interperted as an expanding universe. Stars emit a continuous spectrum of light. However, when this light passes through the cooler outer regions of the star, the elements there absorb some of the frequencies resulting in what are known as Fraunhaufer black lines on the continium. Our sun is used as a standard. Starlight from all sources than what is known as the local group have their sets of dark lines shifted to the red or higher wavelengths. This shift has been interperted as proof that the source and the earth are separating. The shift is now used to determine distance as well. The problem is that using the distance given by the red shift results in at least one source expanding at 10 times the speed of light which is obviously wrong according to the theory of relatively. This leads one to the obvious conclusions that not all the red shift observed can be attributed to the expansion effect. But, note the error. To bring this source into a reasonable rate of expansion means finding other reasons for about 90% of the red shift. This simple but fundamental flaw should be embarrassing to cosmology theorists but it doesn't seem to bother anyone. They prefer to overlook this bit of data as being too mundane to worry about. A lot of other holes in theories and other inabilities are also glossed over.

Similar inabilities exist in other areas of cosmology as well as other scientific theories. Some of this is because of the reduction of all theories to mathematically formulas. Many insisted that these be linear whereas nature generally is not that accomodating. Even our mathematics isn't perfect. The present explaination of the square root of a negartive number being 'imagionary' leaves much to be desired. In calculu one has to add an intergration constant everytime one does an intergration because of 'missing information'. Newton's theories could be applied as a first approximation universally to all things. Einstein's theory of relativety was to fine tune Newton's gravity theory but can no longer be used unversally. It tends t ignore facts that disprove it.

Quantum theory mathematics is probabilistic. The theory calculates the end results but can says nothing about how it gets there which to me means the theory is incomplete. This inability in the math allows for some weird explainations. Black holes which are black because they are so massive and have so much gravity that not even light can escape all at once can evaporate when the 3rd law of theormdynamics is applied which implies that evaporation is stronger than gravity--that is ridiculous. The problem is that it is so easy to creat black holes that the universe should be over run with them which is not the case so something had to be done to limit the number remaining.

The PC is super enough to allow for modeling but still is limited. Many of the models being promoted at the present time have to be limited in scope and the number of variables used. Many of the weather models could not predict what would happen with a Mt. Pititubo explosion. The same applies to what might happen to a WWIII type nuclear war 'winter'. Earthquake models also still are limited in what they can do.

What does all this mean? All theories must be reconciled to facts while models admit their simplifying assumptions. Results of models should be presented as supositions not facts. Let's not get sloppy with our sceince.

"About the blind employment ad we ran—all of our employees answered it' ".

2068 Code --#7

We continue our discussion of routines from the game Conquest. Up to this point we have discussed in:

1. Game background music
2. Sound effects.
3. General Input routine.
4. General Menu routine.
5. Printing of a double wrapping map.
6. Printing of Graph coordinates.

We continue with a map cursor. Whenever one uses a joystick (or for that matter a mouse) to direct the movement of various units one needs a cursor to indicate the desired direction. We use a capitol X for the cursor symbol.

We need to store the screen coordinates of the cursor in two variables called Curx and Cury. The cursor should not run off the map or be allowed to run over the map coordinates which are down the left side and line 22 acrossed the bottom. Therefore, Curx can be from 1 to 31 (in Basic notation) and Cury from 0 to 21.

The input from this routine not only moves the cursor but also nust determine when the joystick button has been pushed as well as interpert all the keyboard commands. Since the keyboard is interperted differently for the game than in map drawing, 2 different print cursor routines are necessary. The simpliest of the two is the one for the map editor which is given below and found in Conquest at 61018.

```
61018 62,1 PRCURS    LD A,1
 :INITILIZE CURX AND CURY
61020 50,171,204     LD (CURX),A
61023 50,170,204     LD (CURY),A
61026 205,63,236     CALL PRMAPP
61029 237,75,179,204 PRCUR LD BC,
(CURY) :THIS IS THE RETURN FOR REPEATS
OF THE ROUTINE IN CASE THE INPUT WAS
MERELY A MOVE OF THE CURSOR.
61033 205,240,238    CALL PIXLOC
 :TRANSLATE THE X AND Y INTO THE FIRST
PIXEL OF THE SCREEN FILE.
61036 235            EX DE,HL
61037 62,88          LD A,"X"
61039 205,18,239     CALL PRCHA
 :PRINT THE CURSOR
```

```
61042 205,255,238    CALL ATTR
61045 126            LD A,(HL)
61046 230,248        AND 248
 :SAVE THE PAPER COLOR
61048 198,7          ADD A,7
 :MAKE THE INK WHITE.
61050 119            LD (HL),A
61051 205,184,235    CALL GETINP (SEE
2068 CODE BYTES #2 FOR THE ROUTINE.)
61054 203,119        BIT 6,A
61056 32,39          JR NZ,KEYBOARD
 :A KEY WAS PRESSED.
61058 203,127        BIT 7,A
61060 40,5           JR Z,MOVE
61062 205,130,239    CALL DOIT
61065 24,218         JR PRCUR
61067 229     MOVE   PUSH HL
 :OLD ATTR ADDR
61068 213            PUSH DE
 :OLD SCR ADDR
61069 197            PUSH BC
 :OLD CUR POSN
61070 205,48,239     CALL INTJOY
:CHANGE BC TO NEW POSITION OF CURSOR
61073 237,67,170,204 LD (CURY),BC
 :SAVE NEW COORDINATES OF CURSOR
61077 193            POP BC
61078 205,90,239     CALL TRANS
61081 205,26,236     CALL PMO
61084 126            LD A,(HL)
 :ERASE OLD CURSOR
61085 209            POP DE
 :REPRINT OLD POSITION. SEE #5 AND #6
FOR ROUTINE.
61086 205,119,237    CALL PRUDG
61089 126            LD A,(HL)
61090 205,254,236    CALL FINDAT
61093 225            POP HL
61094 119            LD (HL),A
61095 24,189         JR PRCUR
61097 254,99 KB      CP 99 "c"
61099 32,5           JR NZ,KBF
61101 205,167,240    CALL CENCUR
 :IN EDIT MODE, IT CENTERS MAP ON THE
CURSOR. IN GAME MODE, ON THE UNIT TO
BE MOVED.
61104 24,179         JR PRCUR
61106 254,102   KBF  CP 102 "f"
61108 32,5           JR NZ,KBG
 :F,G,T AND V ARE THE 4 SCROLLS
61110 205,35,238     CALL SCROLL
61113 24,170         JR PRCUR
61115 254,103   KBG  CP 103 "g"
61117 32,5           JR NZ,KBI
61119 205,53,238     CALL SCROLR
61122 24,161         JR PRCUR
61124 254,105   KBI  CP 105 "i"
61126 32,5           JR NZ,KBM
61128 205,244,246    CALL IDENTIFY
61131 24,152         JR PRCUR
```

```
61133 254,109   KBM  CP 109 "m"
61135 200            RET Z
 :M AND Q RETURN
61136 254,113        CP 113 "q"
61138 200            RET Z
61139 254,116        CP 116 "t"
61141 32,5           JR NZ,KBV
61143 205,78,238     CALL SCROLU
61146 24,137         JR PRCUR
61148 254,118   KBV  CP 118 "v"
61150 32,5           JR NZ,KBA
61152 205,63,238     CALL SCROLD
61155 24,128         JR PRCUR
61157 254,97    KBA  CP 97 "a"
61159 194,101,238    JP PRCUR
 :IF MORE KEYS NEED TO BE READ THEY
WOULD BE INSERTED HERE.
61162 205,120,240    CALL WORLD
61165 195,101,238    JP PRCUR

 :BC = COORDINATES FOR PIXLOC & ATTR
61168 120 PIXLOC     LD A,B
61169 230,248        AND 248
61171 198,64         ADD A,64
61173 103            LD H,A
61174 120            LD A,B
61175 230,7          AND 7
61177 15             RRCA
61178 15             RRCA
61179 15             RRCA
61180 129            ADD A,C
61181 111            LD L,A
 :HL NOW SCREEN FILE ADDRESS OF BC
COORDINATES.
61182 201            RET

61183 120 ATTR       LD A,B
61184 203,47         SRA A
61186 203,47         SRA A
61188 203,47         SRA A
61190 198,88         ADD A,88
61192 103            LD H,A
61193 120            LD A,B
61194 230,7          AND 7
61196 15             RRCA
61197 15             RRCA
61198 15             RRCA
61199 129            ADD A,C
61200 111            LD L,A
 :HL NOW ATTR ADDR OF BC
61201 201            RET

 :DE = SCREEN POSN
 : A = CHARACTER NUMBER
61202 197   PRACHA   PUSH BC
61203 229            PUSH HL
61204 213            PUSH DE
61205 38,0           LD H,0
61207 167            AND A
61208 23             RLA
```

```
61209 23              RLA
61210 203,20          RL H
61212 23              RLA
61213 203,20          RL H
61215 111             LD L,A
61216 124             LD A,H
61217 198,60          ADD A,60
61219 103             LD H,A
 :HL NOW AT PIXELS FOR CHARACTER
61220 6,8             LD B,8
61222 126       PRC   LD A,(HL)
61223 18              LD (DE),A
61224 35              INC HL
61225 20              INC D
61226 16,250          DJNZ PRC
61228 209             POP DE
61229 225             POP HL
61230 193             POP BC
61231 201             RET

 :A = REVERSED BITS FROM JOYSTICK
  BIT 3= R, 2=L, 1=D, 0=U
61232 203,71  INTJOY  BIT 0,A
61234 40,1            JR Z, IJ1
61236 5               DEC B
61237 203,79    IJ1   BIT 1,A
61239 40,1            JR Z,IJ2
61241 4               INC B
61242 203,87    IJ2   BIT 2,A
61244 40,1            JR Z,IJ3
61246 13              DEC C
61247 203,95    IJ3   BIT 3,A
61249 40,1            JR Z,IJ4
61251 12              INC C
 :NOW CHECK BC TO MAKE SURE NOT OFF
SCREEN AS DISCUSSED ABOVE.
61252 120       IJ4   LD A,B
61253 254,255         CP 255
61255 32,1            JR NZ,IJ5
61257 5               INC B
61258 254,22    IJ5   CP 22
61260 32,1            JR NZ,IJ6
61262 4               DEC B
61263 121       IJ6   LD A,C
61264 254,0           CP 0
61266 32,1            JR NZ,IJ7
61268 12              INC C
61269 254,32    IJ7   CP 32
61271 192             RET NZ
61272 13              DEC C
61273 201             RET

 : BC = CURSOR COORDINATES
 THIS ROUTINE CONVERTS THE CURSOR
COORDINATES INTO THE MAP COORDINATES
DEPENDING UPON WHAT IS BEING SHOWN ON
THE SCREEN
61274 58,172,204 TRANS LD A,(MAPCOY)
61277 129             ADD A,C
61278 254,65          CP 65
```

```
61280 56,2                JR C,TR
61282 214,64              SUB 64
61284 79             TR   LD C,A
61285 58,173,204          LD A,(MAPCOX)
61288 128                 ADD A,B
61289 254,65              CP 65
61291 56,2                JR C, TR1
61293 214,64              SUB 64
61295 71                  LD B,A
 :BC NOW IN MAPCOORDINATES
61296 201                 RET

61297 205,90,239 PUTX CALL TRANS
61300 205,26,236          CALL PMO
61303 58,26,236           LD A,(DO)
61306 205,7,241           CALL SETST
61309 216                 RET C
61310 119                 LD (HL),A
61311 195,63,236          JP PRMAPP

 :THE VARIABLE DO MUST BE SET BEFORE
CALLING THIS ROUTINE
 :1= PUT SEA, 2= PUT LAND, 3= PUT
MOUNTAIN, 4= PUT SHOAL, 5= PUT CITY,
6=BLOTCH SEA, 7= BLOTCH LAND, 8= GROW
SEA, 9= GROW LAND
61314 58,164,204 DOIT LD A,(DO)
61317 254,0               CP 0
61319 200                 RET Z
61320 17,158,239          LD DE,TABLE
61323 254,6               CP 6
61325 56,9                JR C,D1
61327 214,5               SUB 5
61329 38,0                LD H,0
61331 167                 AND A
61332 23                  RLA
61333 111                 LD L,A
61334 25                  ADD HL,DE
61335 235            D1   EX DE,HL
61336 26                  LD A,(DE)
61337 111                 LD L,A
61338 19                  INC DE
61339 26                  LD A,(DE)
61340 103                 LD H,A
61341 233                 JP (HL)
.I have always wanted to do a JP (HL)
and here it is. It works great for
lists.
61342 113,239 TABLE PUTX
61344 168,239             BLOTCH
61346 168,239             BLOTCH
61348 218,240             GROW
61350 218,240             GROW
```

We already are 1 column over our normal limit of 4, so will finish th routine with the scrolls next issue.

SMUG BYTES
2461 S. 79th St.
West Allis, WI 53219

FIRST CLASS MAIL

ZXIR CLIVE ALIVE
335 W NEWPORT RD
HOFFMAN ESTATES, IL 60195